UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/657,946	09/09/2003	John M. Brown	10972005-2	2999

7590 08/08/2008

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P. O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
CHOW, JEFFREY J	

ART UNIT	PAPER NUMBER
2628	

MAIL DATE	DELIVERY MODE
08/08/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/657,946	BROWN ET AL.	
	Examiner	Art Unit	
	Jeffrey J. Chow	2628	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _09 September 2003_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _21-71 and 73-89_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _21-26,30-33,38-42,44,49-57,61-63,65-70,73-76,78-85,88 and 89_ is/are rejected.

7)☒ Claim(s) _27-29,34-37,43,45-48,58-60,64,71,77,86 and 87_ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Response to Arguments

Applicant's arguments claims filed 19 May 2008, have been fully considered and are persuasive. Therefore, the rejection has been withdrawn. However, upon further consideration, a new ground(s) of rejection is made.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 21 – 25, 30 – 33, 38 – 42, 44, 49 – 57, 61 – 63, 65 – 69, 73 – 76, 78 – 85, 88, and 89 are rejected under 35 U.S.C. 103(a) as being unpatentable over Scully et al. (US 5,982,399) in view of Murphy (US 6,348,919) and Krech, Jr. (US 6,057,852) and Applicant's Admitted Prior Art (page 20, lines 21 – 24, lines 26 – 27).

Regarding independent claim 21, Scully teaches a graphics system including a graphics application (column 3, line 22: graphics application 118) communicating with graphics hardware (Figure 3: High-Speed Graphics Hardware 132, Low-Cost Frame Buffer 134, and Printer 110) through a first driver (Figure 3: Renderer 3 126; Figures 4 and 5: Software Z-Buffer Renderer 140) sharing an interface (Figure 3: Drawing Interface 116) with the graphics application (Figure 3: Render 3 126 connected with Graphics Application 118) and a second driver (Figure 3: Renderer 1 130; Figures 4 and 5: Physical Render 136) sharing another

interface with the graphics hardware (Figure 3: Renderer 1 130 sharing connections between High-Speed Graphics Hardware 132, Low-Cost Frame Buffer 134, and Printer 110; column 3, lines 39 – 45: the graphics hardware are bounded by renderer stack having multiple renderers), a graphics interface (column 6, lines 15 – 23: more renders can be added to the renderer stack), a performance optimization apparatus (Figure 3: Renderer 2 128). Scully did not expressly disclose the first driver transmits original graphics call sequences to the second driver and the performance optimization apparatus. Murphy discloses processing block are connected in a long pipeline with communication with the adjacent blocks being done through message passing (column 8, lines 7 – 13) and when a block receives a message, the block either not recognize the message and passes it on to the next block, recognize it as updating some local state to the block and terminates the message, or recognize the message, work with the message and either send a new message and/or modify the message (column 8, lines 30 - 43) and if a message fails to pass, do not pass the message to another block (column 8, lines 44 – 60). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have renderers that filters through messages of either passing messages along or not passing messages along, especially not passing messages along that fails a test. One would be motivated to do so because this would allow the system to be efficient by not passing through non-useful messages along the pipeline.

Scully did not expressly disclose the performance optimization apparatus comprises at least one graphics call sequence optimizer, operationally interposed between the first and second driver, configured to process one of the original graphics call sequence to produce an optimized graphics call sequence, however Scully does disclose Renderer 2 128 is between Renderer 1 130

and Render 3 126 (Figure 3) each render being able to modify commands (column 3, line 46 – 55 and Figure 2). Krech, Jr. discloses a identification of several consecutive graphics primitives, or vertices, of equal color, the system would enter into a constant color mode (column 9, lines 21 – 45) with an example of vertex 0 is red and vertex 1 – 6 are all green (column 10, lines 60 – 65) a simplified/optimized code is illustrated in column 11, lines 1 – 16. It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate a renderer or modifying Renderer 2 to optimize code having constant color, as taught by Krech, Jr. One would be motivated to do so because this would be efficient and faster to produce an image.

Scully did not explicitly disclose the performance optimization apparatus comprises measuring means for measuring performance of the graphics system when processing the optimized graphics call sequence. APA teaches measuring means for measuring performance of an optimized graphics call sequence (page 20, lines 21-24, lines 26-27). It would be obvious to one skilled in the art at the time of the invention to combine the teachings of Scully with the teachings of APA in order to evaluate the performance of graphic calls. One would be motivated to do so because this allows users to evaluate benchmark tests.

Regarding dependent claim 22, Scully teaches the optimized graphics call sequence is provided to the graphics hardware (Figure 2: Modify Command Stream 122 then Transmit Command Stream To Sink 124; column 3, lines 64 – 67: high-speed graphics hardware 132; combination of Murphy would allow the modified command to pass through Renderer 1 130).

Regarding dependent claim 23, Scully teaches at least one graphics call sequence optimizer comprising

a first graphics call sequence optimizer that receives the original graphics call sequence and generates an intermediate optimized graphics call sequence (column 3, lines 56 – 67: A graphics application 118 provides commands to the renderer stack. Each renderer 126, 128, and 130 in the stack receives drawing commands from a source, e.g., the renderer above it, and sends drawing commands to a sink, e.g., the renderer below it; Figure 2: Modify Command Stream 122; column 3, line 46 – 55: modified command), and

a second graphics call sequence optimizer that receives the intermediate optimized graphics call sequence and generates the optimized graphics call sequence (column 3, lines 56 – 67: A graphics application 118 provides commands to the renderer stack. Each renderer 126, 128, and 130 in the stack receives drawing commands from a source, e.g., the renderer above it, and sends drawing commands to a sink, e.g., the renderer below it; Figure 2: Modify Command Stream 122; column 3, line 46 – 55: modified command).

Regarding dependent claim 24, Scully teaches a capture mechanism configured to capture the original graphics call sequence transmitted between the first and second drivers (Figure 2: Receive Command Stream From Source 120 and Modify Command Stream 122) and to provide the capture graphics call sequence to the at least one optimization means (Figure 2: Transmit command Stream To Sink 124).

Regarding dependent claim 25, Scully did not expressly disclose at least one graphics call sequence optimizer comprises a graphics state call coalescer constructed and arranged to eliminate from a continuous series of graphics state calls in the original graphics call sequence one or more graphics state calls that do not cause the continuous series of graphics state calls to effect a change in a rendering by the graphics hardware. Murphy discloses processing block are connected in a long pipeline with communication with the adjacent blocks being done through message passing (column 8, lines 7 – 13) and when a block receives a message, the block either not recognize the message and passes it on to the next block, recognize it as updating some local state to the block and terminates the message, or recognize the message, work with the message and either send a new message and/or modify the message (column 8, lines 30 - 43) and if a message fails to pass, do not pass the message to another block (column 8, lines 44 – 60). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have renderers that filters through messages of either passing messages along or not passing messages along, especially not passing messages along that fails a test. One would be motivated to do so because this would allow the system to be efficient by not passing through non-useful messages along the pipeline.

Regarding dependent claim 30, Scully teaches at least the apparatus is implemented in the second driver to cause the second driver to generate the optimized graphics call sequence during real-time operations of the graphics system (column 3, lines 39 – 45: the graphics hardware are bounded by renderer stack having multiple renderers 116; Figure 2: Modify Command Stream 122 then Transmit Command Stream To Sink 124). Scully did not expressly disclose receiving

the original graphics call sequence from the first driver. Murphy discloses processing block are connected in a long pipeline with communication with the adjacent blocks being done through message passing (column 8, lines 7 – 13) and when a block receives a message, the block either not recognize the message and passes it on to the next block, recognize it as updating some local state to the block and terminates the message, or recognize the message, work with the message and either send a new message and/or modify the message (column 8, lines 30 - 43) and if a message fails to pass, do not pass the message to another block (column 8, lines 44 – 60). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have renderers to pass message through if the message is not appropriate for the renderer and to modify the message if the message is appropriate to the renderer. One would be motivated to do so because this would allow the renderers to be efficient in only processing messages pertaining to that particular renderer, wherein specialized renderers are faster and efficient in specific processes, encapsulated in a message.

Regarding dependent claim 31, Scully teaches at least one optimization means is implemented in the graphics hardware to cause the graphics hardware to generate optimized graphics call sequence during real-time operations of the graphics system (column 3, lines 39 – 45: the graphics hardware are bounded by renderer stack having multiple renderers 116; Figure 2: Modify Command Stream 122 then Transmit Command Stream To Sink 124). Scully did not expressly disclose in response to receiving the original graphics call sequence from the second driver. Murphy discloses processing block are connected in a long pipeline with communication with the adjacent blocks being done through message passing (column 8, lines 7 – 13) and when

a block receives a message, the block either not recognize the message and passes it on to the next block, recognize it as updating some local state to the block and terminates the message, or recognize the message, work with the message and either send a new message and/or modify the message (column 8, lines 30 - 43). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have renderers that filters through messages of either passing messages along or not passing messages along, especially not passing messages along that fails a test. One would be motivated to do so because this would allow the system to be efficient by not passing through non-useful messages along the pipeline. Scully teaches to generate the optimized graphics call sequence during real-time operations of the graphics system (Figure 2: Modify Command Stream 122 then Transmit Command Stream To Sink 124).

Regarding dependent claim 32, Scully teaches at least one optimization means is distributed in the second driver and the graphics hardware to generate the optimized graphics call sequence during real-time operations of the graphics system (Figure 2: Modify Command Stream 122 then Transmit Command Stream To Sink 124; column 3, lines 64 – 67: high-speed graphics hardware 132; combination of Murphy would allow the modified command to pass through Renderer 1 130).

Regarding dependent claim 33, Scully did not expressly disclose the plurality of graphics vertex calls in the coalesced primitive command set define a strip primitive and wherein the graphics primitive coalescer is constructed and arranged to remove redundant graphics vertex

calls from the coalesced primitive command set that define vertices common to neighboring primitives of the strip primitive and to alter a primitive type specified by the coalesced primitive command set to identify the primitive strip rather than the discrete primitive type. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate well known OpenGL™ functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 38, Scully did not expressly disclose at least one graphics call sequence optimizer comprises a vertex loop generator constructed and arranged to generate a repeatable loop to efficiently process repetitive series of graphics calls occurring in a primitive command set in the original graphics call sequence. Krech, Jr. discloses repeating the steps of incrementing the count and operating on primitive elements until a specified count has been reached (column 3, lines 46 – 64). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have a loop counter and to terminate the loop once the count has reached a limit. One would be motivated to do so because this would prevent overflow.

Regarding dependent claim 39, Scully did not expressly disclose means for measuring performance of the original graphics call sequence to obtain a performance baseline against which the performance of the optimized graphics call sequence can be compared. APA teaches the measuring means measures the performance of a graphic call sequence to obtain a performance baseline against to compare the performance of other graphic call sequences (pg. 20, lines 19-27). It would be obvious to one skilled in the art at the time of the invention to combine the teachings of Scully with the teachings of APA in order to evaluate the performance of graphic calls. One would be motivated to do so because this allows users to evaluate benchmark tests.

Regarding dependent claim 40, Scully did not expressly disclose measuring means comprises means for compiling the original and the optimized graphics call sequences. APA teaches the measuring means for compiling a graphic call sequence (pg. 20, lines 21-24, lines 26-27). It would be obvious that both call sequences are compiled for performance evaluating. One would be motivated to do so because this allows users to evaluate benchmark tests.

Regarding independent claim 41, Scully teaches in a graphics system including a graphics application (column 3, line 22: graphics application 118) communicating with graphics hardware (Figure 3: High-Speed Graphics Hardware 132, Low-Cost Frame Buffer 134, and Printer 110) through a first driver (Figure 3: Renderer 3 126; Figures 4 and 5: Software Z-Buffer Renderer 140) interfaced (Figure 3: Drawing Interface 116) with the graphics application (Figure 3: Render 3 126 connected with Graphics Application 118) and a second driver (Figure 3:

Renderer 1 130; Figures 4 and 5: Physical Render 136) interfaced with the graphics hardware (Figure 3: Renderer 1 130 sharing connections between High-Speed Graphics Hardware 132, Low-Cost Frame Buffer 134, and Printer 110; column 3, lines 39 – 45: the graphics hardware are bounded by renderer stack having multiple renderers), and a graphics interface (column 6, lines 15 – 23: more renders can be added to the renderer stack). Scully did not expressly disclose the first driver transmits original graphics call sequences to the second driver. Murphy discloses processing block are connected in a long pipeline with communication with the adjacent blocks being done through message passing (column 8, lines 7 – 13) and when a block receives a message, the block either not recognize the message and passes it on to the next block, recognize it as updating some local state to the block and terminates the message, or recognize the message, work with the message and either send a new message and/or modify the message (column 8, lines 30 - 43) and if a message fails to pass, do not pass the message to another block (column 8, lines 44 – 60). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have renderers that filters through messages of either passing messages along or not passing messages along, especially not passing messages along that fails a test. One would be motivated to do so because this would allow the system to be efficient by not passing through non-useful messages along the pipeline. Scully teaches a method for optimizing the original graphics call sequence to generate an optimized graphics call sequence causing the graphics hardware to render with improved performance (column 3, line 46 – 55: modified command), a same image as the original graphics call sequence, the method comprising

capturing the original graphics call sequence (Figure 2: Receive Command Stream From Source 120).

Scully did not expressly disclose restructuring the original graphics call sequence to produce the optimized graphics call sequence, however Scully does disclose Renderer 2 128 is between Renderer 1 130 and Render 3 126 (Figure 3) each render being able to modify commands (column 3, line 46 – 55 and Figure 2). Krech, Jr. discloses a identification of several consecutive graphics primitives, or vertices, of equal color, the system would enter into a constant color mode (column 9, lines 21 – 45) with an example of vertex 0 is red and vertex 1 – 6 are all green (column 10, lines 60 – 65) a simplified/optimized code is illustrated in column 11, lines 1 – 16. It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate a renderer or modifying Renderer 2 to optimize code having constant color, as taught by Krech, Jr. One would be motivated to do so because this would be efficient and faster to produce an image.

Scully did not expressly disclose measuring performance of the graphics system when processing the optimized graphics call sequence. APA teaches measuring means for measuring performance of an optimized graphics call sequence (page 20, lines 21-24, lines 26-27). It would be obvious to one skilled in the art at the time of the invention to combine the teachings of Scully with the teachings of APA in order to evaluate the performance of graphic calls.

Regarding dependent claim 42, Scully did not expressly disclose restructuring the original graphics call sequence comprises removing from a continuous series of graphics state calls in the original graphics state sequence all redundant, duplicative and otherwise unnecessary

graphics state calls to form a coalesced continuous series of graphics state calls, wherein the coalesced continuous series of graphics state calls includes a number of graphics state calls that is less than a number of graphics state calls in the continuous series of graphics state calls. Krech, Jr. discloses well known graphic primitives, such as a “triangle strip” that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully’s system to incorporate well known OpenGL™ well known functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 44, Scully did not expressly disclose restructuring the original graphics call sequence comprises coalescing graphics vertex calls contained in a continuous series of primitive command sets that render primitives of the same type in the original graphics call sequence to generate a corresponding coalesced primitive command set in the optimized graphics call sequence effecting a same rendering in the graphics hardware as the original graphics call sequence. Krech, Jr. discloses well known graphic primitives, such as a “triangle strip” that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully’s system to incorporate OpenGL™ well known functions that optimizes creation of at least triangle strip,

which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 49, Scully did not expressly restructuring the original graphics call sequence comprises converting, prior to compiling a continuous series of primitive command sets, the specified primitive type from a primitive type that cannot be coalesced to a primitive type that can be coalesced. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle, wherein the triangle contains 3 vertices (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate OpenGL™ functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 50, Scully did not expressly disclose the graphics API is an OpenGL graphics API, and wherein the graphics primitives comprise line strips, triangle strips, quadrilateral strips, triangle fans, line loops and polygon primitive types. Krech, Jr. discloses line strip, triangle strip, quad mesh, triangle fan, line loop, and quad (Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate well known OpenGL™ functions that optimizes creation of primitives. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 51, Krech, Jr. teaches the graphics API is an OpenGL API, and wherein converting the specified primitive type comprises converting a primitive type specified in a primitive command set from a line strip to a line when the primitive command set includes 2 graphics vertex calls (Table 1).

Regarding dependent claim 52, Krech, Jr. teaches the graphics interface is an OpenGL API, and wherein converting the specified primitive type comprises converting a primitive type specified in a primitive command set from a triangle strip to a triangle when the primitive command set includes 3 graphics vertex calls (Table 1).

Regarding dependent claim 53, Krech, Jr. teaches the graphics interface is an OpenGL API, and wherein converting the specified primitive type comprises converting a primitive type specified in a primitive command set from a quadrilateral strip to a quadrilateral when the primitive command set includes 4 graphics vertex calls (Table 1).

Regarding dependent claim 54, Krech, Jr. teaches the graphics interface is an OpenGL API, and wherein converting the specified primitive type comprises converting a primitive type specified in a primitive command set from a triangle fan to a triangle when the primitive command set includes 3 graphics vertex calls (Table 1).

Regarding dependent claim 55, Krech, Jr. teaches the graphics interface is an OpenGL API, and wherein converting the specified primitive type comprises converting a primitive type specified in a primitive command set from a line loop to a line when the primitive command set includes 2 graphics vertex calls (Table 1).

Regarding dependent claim 56, Krech, Jr. teaches the graphics interface is an OpenGL API, and wherein converting the specified primitive type comprises converting a primitive type specified in a primitive command set from a polygon strip to a quadrilateral when the primitive command set includes 4 graphics vertex calls (Table 1).

Regarding dependent claim 57, Scully did not expressly disclose restructuring the original graphics call sequence comprises creating a vertex array having vertices identified in a series of graphics vertex calls of a primitive command set for reference by an associated pointer and graphics array call to render one or more of a specified primitive. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate OpenGL™ well known functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 61, Scully did not expressly disclose restructuring the captured graphics call sequence comprises creating a vertex array having vertices identified in a series of graphics vertex calls of a primitive command set in the original graphics call sequence for reference by an associated graphics array call to render one or more specified primitives. Krech, Jr. discloses well known graphic primitives, such as a “triangle strip” that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully’s system to incorporate OpenGL™ well known functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 62, Scully did not expressly disclose compiling a continuous series of primitive command sets comprises identifying graphics calls that comprise a repetitive series, counting a number of occurrences each the graphics call continuously appears in the primitive command set, and generating an optimized graphics call sequence effecting a loop execution of the repetitive graphics calls for the identified number of times the particular repetitive pattern should be implemented. Krech, Jr. discloses repeating the steps of incrementing the count and operating on primitive elements until a specified count has been reached (column 3, lines 46 – 64). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have a loop counter and to terminate the

loop once the count has reached a limit. One would be motivated to do so because this would prevent overflow.

Regarding dependent claim 63, Scully did not expressly disclose restructuring the original graphics call sequence comprises removing from a continuous series of graphics state calls in the original graphics state sequence all redundant, duplicative and otherwise unnecessary graphics state calls to form a coalesced continuous series of graphics state calls, wherein the coalesced continuous series of graphics state calls includes a number of graphics state calls less than a number of graphics state calls in the continuous series of graphics state calls and causing the graphics hardware to render a same image as the continuous series of graphics state calls. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate OpenGL™ well known functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claims 65 and 66, claims 65 and 66 are similar in scope as to claims 39 and 40, thus the rejections for claims 39 and 40 hereinabove is applicable to claims 65 and 66.

Regarding independent claim 67, Scully teaches in a graphics system including a graphics application (column 3, line 22: graphics application 118) communicating with graphics hardware (Figure 3: High-Speed Graphics Hardware 132, Low-Cost Frame Buffer 134, and Printer 110) through a first driver (Figure 3: Renderer 3 126; Figures 4 and 5: Software Z-Buffer Renderer 140) interfaced (Figure 3: Drawing Interface 116) with the graphics application (Figure 3: Render 3 126 connected with Graphics Application 118) and a second driver (Figure 3: Renderer 1 130; Figures 4 and 5: Physical Render 136) interfaced with the graphics hardware (Figure 3: Renderer 1 130 sharing connections between High-Speed Graphics Hardware 132, Low-Cost Frame Buffer 134, and Printer 110; column 3, lines 39 – 45: the graphics hardware are bounded by renderer stack having multiple renderers), the drivers communicating with each other through a graphics interface (Figure 3: Drawing Interface 116).

Scully did not expressly disclose a performance optimization apparatus comprising graphics call sequence optimization means for processing the original graphics call sequence to produce an optimized graphics call sequence, however Scully does disclose Renderer 2 128 is between Renderer 1 130 and Render 3 126 (Figure 3) each render being able to modify commands (column 3, line 46 – 55 and Figure 2). Krech, Jr. discloses a identification of several consecutive graphics primitives, or vertices, of equal color, the system would enter into a constant color mode (column 9, lines 21 – 45) with an example of vertex 0 is red and vertex 1 – 6 are all green (column 10, lines 60 – 65) a simplified/optimized code is illustrated in column 11, lines 1 – 16. It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate a renderer or modifying Renderer 2 to

optimize code having constant color, as taught by Krech, Jr. One would be motivated to do so because this would be efficient and faster to produce an image.

Scully did not expressly disclose a performance optimization apparatus comprising measuring means for measuring performance of the graphics system when processing the optimized graphics call sequence. APA teaches measuring means for measuring performance of an optimized graphics call sequence (page 20, lines 21-24, lines 26-27). It would be obvious to one skilled in the art at the time of the invention to combine the teachings of Scully with the teachings of APA in order to evaluate the performance of graphic calls.

Regarding dependent claim 68, Scully teaches a capture means, operatively coupled to a communications path coupling the first and second drivers (column 6, lines 15 – 23: more renders can be added to the renderer stack), for capturing the original graphics call sequence transmitted between the first and second drivers (Figure 2: Receive Command Stream From Source 120 and Modify Command Stream 122), and means for providing the captured graphics call sequence to the at least one graphics call sequence optimization means (Figure 3: Renderer 2 128 is between Renderer 1 130 and Render 3 126).

Regarding claim 69, claim 69 is similar in scope as to claim 25, thus the rejection for claim 25 hereinabove is applicable to claim 69.

Regarding dependent claim 73, Scully teaches the optimizer is implemented in the first driver (column 10, lines 22 – 25: version renderer 182).

Regarding dependent claim 74, Scully teaches the optimizer is implemented in the second driver (column 4, line 47: physical renderer 136).

Regarding dependent claim 75, Scully teaches the optimization means is implemented in the graphics hardware (column 3, lines 39 – 45: the graphics hardware and renderer stack containing renderers).

Regarding dependent claim 76, Scully did not expressly disclose the plurality of graphics vertex calls in the coalesced primitive command set define a strip primitive and wherein the graphics primitive coalescing means comprises means for removing redundant graphics vertex calls from the coalesced primitive command set that define vertices common to neighboring primitives of the strip primitive and to alter a primitive type specified by the coalesced primitive command set to identify the primitive strip rather than the discrete primitive type. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle, wherein the triangle contains 3 vertices (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate OpenGL™ functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 78, Scully did not expressly disclose the optimization means further comprises primitive type converting means for converting a primitive type specified in a primitive command set in the original graphics call sequence from a non-combinable primitive type including primitives that cannot be coalesced by the graphics primitive coalescing means to a combinable primitive type that can be coalesced by the graphics primitive coalescing means. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that creates a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle, wherein the triangle contains 3 vertices (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate OpenGL™ functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 79, Scully did not expressly disclose the optimization means comprises vertex array generating means for creating a vertex array having vertices identified in a series of graphics vertex calls of a primitive command set in the original graphics call sequence, and to generate an associated offset for reference by a graphics array call that uses the array of vertices to render a number of specified primitives. Krech, Jr. discloses well known graphic primitives, such as a "triangle strip" that n vertices-draws a triangle after the first 3 vertices then another triangle for each additional vertex, the last 2 vertices sent are combined with the next vertex sent to from the 3 vertices of the triangle and draws n - 2 triangles, wherein

the triangle contains 3 vertices (column 9, lines 46 – 53 and Table 1). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to incorporate OpenGL™ functions that optimizes creation of at least triangle strip, which removes redundant vertices. One would be motivated to do so because this would be faster and efficient.

Regarding dependent claim 80, Scully did not expressly disclose the optimization means comprises vertex loop generating means for generating a repeatable loop to efficiently process repetitive series of graphics calls occurring in a primitive command set in the original graphics call sequence. Krech, Jr. discloses repeating the steps of incrementing the count and operating on primitive elements until a specified count has been reached (column 3, lines 46 – 64). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have a loop counter and to terminate the loop once the count has reached a limit. One would be motivated to do so because this would prevent overflow.

Regarding dependent claims 81 and 82, claims 81 and 82 are similar in scope as to claims 39 and 40, thus the rejections for claims 39 and 40 hereinabove is applicable to claims 81 and 82.

Regarding independent claim 83, claim 83 is similar in scope as to claim 41, thus the rejection for claim 41 hereinabove is applicable to claim 83.

Regarding dependent claims 84 and 85, claims 84 and 85 are similar in scope as to claims 33, 44, thus the rejections for claims 33, 44 hereinabove are applicable to claims 84 and 85.

Regarding dependent claims 88 and 89, claims 88 and 89 are similar in scope as to claims 39 and 40, thus the rejections for claims 39 and 40 hereinabove is applicable to claims 88 and 89.

Claims 26 and 70 are rejected under 35 U.S.C. 103(a) as being unpatentable over Scully et al. (US 5,982,399) in view of Murphy (US 6,348,919) and Applicant's Admitted Prior Art (page 20, lines 21 – 24, lines 26 – 27) and Krech, Jr. (US 6,057,852) and "OpenGL™ Reference Manual" (pg 300, 1992).

Regarding dependent claim 26, Scully did not expressly disclose at least one graphics call sequence optimizer comprises a graphics primitive coalescer constructed and arranged to coalesce a continuous series of primitive command sets occurring in the original graphics call sequence, the primitive command sets specifying a same type of discrete primitive and comprising at least one graphics vertex call, the graphics primitive coalescer generating a coalesced primitive command set comprising a plurality of graphics vertex calls causing the graphics hardware to generate a same rendering as the continuous series of primitive command sets. Murphy discloses processing block are connected in a long pipeline with communication with the adjacent blocks being done through message passing (column 8, lines 7 – 13) and when a block receives a message, the block either not recognize the message and passes it on to the next block, recognize it as updating some local state to the block and terminates the message, or recognize the message, work with the message and either send a new message and/or modify the

message (column 8, lines 30 - 43) and if a message fails to pass, do not pass the message to another block (column 8, lines 44 – 60). OpenGL™ Reference Manual discloses vertex calls (pg 300). It would have been obvious for one of ordinary skill in the art at the time of the invention to modify Scully's system to have renderers that filters through messages having OpenGL™ commands, such as GLVERTEX, and either passing messages along or not passing messages along, especially not passing messages along that fails a test. One would be motivated to do so because this would allow the system to be efficient by not passing through non-useful messages along the pipeline.

Regarding claim 70, claim 70 is similar in scope as to claim 26, thus the rejection for claim 26 hereinabove is applicable to claim 70.

Allowable Subject Matter

Claims 27 – 29, 34 – 37, 43, 45 – 48, 58 – 60, 64, 71, 77, 86, and 87 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

The following is an examiner's statement of reasons for allowance: upon further considerations, the combination of elements recited in claims 27-29, 34 – 37, 43, 45 – 48, 58 – 60, 64, 71, 77, 86, and 87 are not taught by the cited prior arts.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jeffrey J. Chow whose telephone number is (571)-272-8078. The examiner can normally be reached on Monday - Friday 10:00AM - 5:00PM (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on (571)-272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

JJC

/Ulka Chauhan/

Supervisory Patent Examiner, Art Unit 2628